(19) Europäisches Patentamt
European Patent Office
Office européen des brevets

(11) EP 1 049 307 A1

(12) EUROPEAN PATENT APPLICATION

(43) Date of publication:
02.11.2000 Bulletin 2000/44

(51) Int Cl.[7]: **H04L 29/06**

(21) Application number: **99480027.4**

(22) Date of filing: **29.04.1999**

(84) Designated Contracting States:
AT BE CH CY DE DK ES FI FR GB GR IE IT LI LU MC NL PT SE
Designated Extension States:
AL LT LV MK RO SI

(71) Applicant: **International Business Machines Corporation**
Armonk, N.Y. 10504 (US)

(72) Inventors:
- **Lamberton, Marc**
  06600 Antibes (FR)
- **Montagnon, Eric**
  06510 Gattières (FR)
- **Levy-Abegnoli, Eric**
  06200 Nice (FR)
- **Thubert, Pascal**
  06140 Vence (FR)

(74) Representative: **Etorre, Yves Nicolas**
Compagnie IBM France,
Département Propriété Intellectuelle
06610 La Gaude (FR)

(54) Method and system for dispatching client sessions within a cluster of servers connected to the World Wide Web

(57) A method and system for preserving load balancing of the client transactions, for the whole duration of the client sessions, in a Web site implemented in the form of a cluster of servers is described. The invention manages to send only the initial request of each client session to the site load balancer thus, greatly enhancing the capability of the site to accept new session requests. All subsequent requests from a client are forwarded directly to the server first selected so that the sessions cannot be later broken by the load balancer. The scheme works regardless of the fact that the client is beyond a proxy or a firewall and greatly contributes to the performance of the Web site.

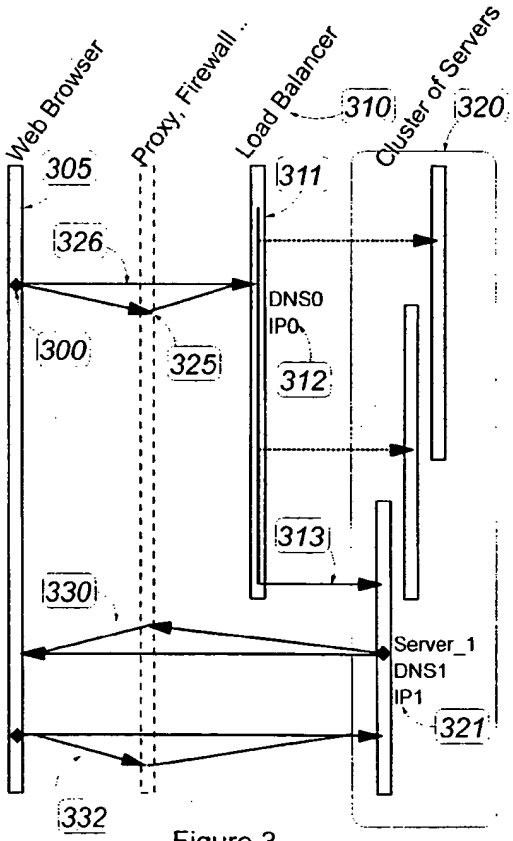

Figure 3

EP 1 049 307 A1

Description

Field of the Invention

[0001] The present invention deals with the global Internet network and more particularly to those of the Internet Servers of World Wide Web (WWW) sites organized as a cluster or group of servers forming a single entity.

Background of the Invention

[0002] The Internet is the world's largest network, and it has become essential in organizations such as government, academia and commercial enterprises. Transactions over the Internet are becoming more common, especially in the commercial arena. The information that organizations have on their traditional or legacy business applications may now be published and accessible to a wide audience. This access may include a person checking a bank savings account, making a hotel reservation or buying tickets for a concert. Making this information or service available for their customers is a competitive advantage for any organization. However, regardless of the innovation and potential benefits provided by a company's Internet solution, its value is greatly reduced if the information cannot be accessed in a reasonable response time.

[0003] The load on an Internet site is unlikely to remain constant. The number of accesses on a Web server can increase for several reasons.

1. Most companies add their Web site's address to television, radio and print advertising and to product catalogues and brochures. Therefore, awareness of the Web site grows.
2. As time passes, the Web site gains better coverage in the on-line search engines.
3. Assuming the site is providing useful information or a useful service to customers, repeat visitors should increase.
4. Most Web sites begin simply, with fairly modest content, mostly text, with some images. As the site designers grow in confidence, more resources are allocated, and as Web users in general increase their modem speeds, most sites move towards richer content. Thus, not only do hit rates increase, but the average data transfer per hit also rises.
5. Most sites begin as presence sites providing corporate visibility on the Internet and making information about the company available to potential customers. Most present sites use predominantly static Hyper Text Marked-up Language or HTML pages. Static pages are generated in advance and stored on disk. The server simply reads the page from the disk and sends it to the browser. However, many companies are now moving towards integration applications that allow users of the Web site to directly access information from the company's existing applications. This could include checking the availability of products, querying bank account balances or searching problem databases. These applications require actual processing on the server system to dynamically generate the Web page. This dramatically increases the processing power required in the server.

[0004] There are several ways to deal with the growth of an Internet site like purchasing an initial system that is much too large. This is one way to deal with Web site growth; however, most companies are not willing to invest large sums of money in a system that is much larger than they require particularly since the benefits that they will gain from the site have yet to be proven. Most prefer to purchase a minimal initial system and to upgrade in the future as the site demonstrates its worth to the company. In this realm of solutions load balancing between multiple servers is very often used. In this case, the load for the overall site is balanced between multiple servers. This allows scaling beyond the maximum performance available from a single system and allows for easy upgrading by simply installing additional servers and reconfiguring the cluster to use the additional servers. This solution can also provide the added benefit of higher server availability. The load-balancing software can automatically allow for the failure of a single server and balance the load between the remaining sites. Because the Internet model allows the distribution of services among different servers, called Internet Servers it is definitively feasible not to tie an application to one specific server. Instead, the service belongs to a group of servers; so an additional computer can be added or removed when necessary. However, grouping the set of servers in a single entity, implies that load balancing is well performed between these servers so as to actually achieve optimum performance. A discussion on this and more on load balancing can be found for example in a paper by Dias et al., "A Scalable and Highly Available Web Server", Digest of Papers, Compcon 1996, Technologies for the Information Superhighway, Forty-first IEEE Computer Society International Conference (Cat. No. 96CB35911), pp. 85-92, Feb. 1996.

[0005] Load-balancing products have made their way to the market. IBM's eNetwork Dispatcher (eND) is one of those products now commercially available. It creates the illusion of having just one server by grouping systems together into a cluster that behaves as a single, virtual server. The service provided is no longer tied to a specific server system; so it is possible to add or remove systems from the cluster, or shutdown systems for maintenance, while maintaining continuous service for the clients. The balanced traffic among servers seems for the end users to be a single, virtual server. The site thus appears as a single IP (Internet Protocol) address to the world. All requests are sent to the IP address of the a Network Dispatcher machine, which de-

cides with each client request which server is the best one to accept requests, according to certain dynamically set weights. Network Dispatcher routes the clients' request to the selected server, and then the server responds directly to the client without any further involvement of eND. This makes it possible to have a small bandwidth network for incoming traffic (like Ethernet or token ring) and a large bandwidth network for outgoing traffic (like ATM - Asynchronous Transfer Mode or FDDI - Fiber Distributed Data Interface or Fast Ethernet). It can also detect a failed server and route traffic around it. General information on the way of performing load balancing between multiple servers and on eND product can be found in a 'Redboook' by IBM published by the Austin, Texas center of the International Technical Support Organization (ITSO) and untitled "Load-Balancing Internet Servers" under the reference SG24-4993 on December 1997.

[0006] Those products are great to achieve what they have been designed for, i.e., load-balancing and indeed allow to build scalable Web site capable of coping with a rapidly growing demand for higher traffic. However, they have created their own difficulties too. Because there are now numerous sophisticated Web servers that allow to handle dynamic Web pages they need to be session-aware for every user accessing their service. Several techniques indeed exist to keep track of the context in which a particular user is accessing a Web server. They are of two kinds:

- the contextual data is circulating, back and forth, in the IP packets exchanged between the client and the servers. For example, it can be part of the Web pages themselves.
- or the contextual data is kept in the Web server active memory or on disk. This second solution is necessary whenever the amount of data needed to define each session context is too large to be practically transported over the network with each transaction between the client and the servers.

[0007] Then, load-balancing products such as eND manage not to dispatch randomly the traffic to the servers of their cluster. They keep track of the user requests which must end up into the same server while a session is active. To achieve this, the usual technique, well known from the art, consists in utilizing the IP address of the client. Then, each transaction coming from the same IP address is dispatched to the same server.

[0008] However, this does not fit in the now frequent situations where the end user and the server are on either side of a proxy, socks or fire-wall. All those devices, part of the Internet, are intended to deal with specific problems like, for example, the isolation of an intranet that must not be freely accessible by outsiders without any control thus, leading to place a fire-wall at the intranet gateway. Or a proxy, so as the users within an intranet are seeing the whole Internet through a common gateway device, somehow caching it, in an attempt to achieve overall better performance. In these situations, *the client IP address is not actually known* by the network dispatcher which establishes in fact a TCP connection (the Transport Control Protocol of the Internet protocol suite) with the proxy, the socks or the fire-wall rather than directly with the end-user. Therefore, the network dispatcher is no longer session aware that is, has no information that would allow it to decide that a particular end-user, for example located beyond a proxy, that was engaged into a transaction such as buying a product from a virtual shop, an application that was first selected by the dispatcher on a particular server in the cluster of servers, has not yet completed. Then, further requests from the end-user, sometimes occuring after a long pause, could be dispatched differently by the network dispatcher just because it does its job of balancing the traffic towards a less busy server within the cluster with the obvious consequence that the new server is not aware of the transaction in progress.

[0009] And there is another undesirable effect of having the end-users beyond a proxy for a load balancer. All the individual users within a group, for example an intranet then, appear to the load balancer as a single user because their IP address is the same since it is the one of the proxy or fire-wall. Therefore, the load balancer which tends to maintain the dispatching of a given user towards the same server, in an attempt not to break sessions, at least while an inactivity timer has not elapsed, keep sending the traffic of the whole intranet to the same server. This seriously goes against what this kind of product is trying to achieve, i.e., load balancing. Although the individual users within a group would certainely enjoy not being served by the sometimes same busy server, because they are seen as being a single client by the load balancer, it is no longer possible to discriminate the individual users.

Object of the Invention

[0010] Thus, it is a broad object of the invention to overcome the shortcomings, as noted above, of the prior art and therefore enabling a particular server, within a cluster of servers, to continue serving a given end-user while the current session is active and being able to discriminate the individual users within a group (intranet) so as to maintain a good load balancing over the cluster of servers.

[0011] It is a further object of the invention to improve the efficiency of the load balancer by requiring only one interrogation per session thus, freeing it to dispatch even more transactions over the cluster of servers.

[0012] Further advantages of the present invention will become apparent to the ones skilled in the art upon examination of the drawings and detailed description. It is intended that any additional advantages be incorporated herein.

Summary of the Invention

[0013] A method and system for preserving load balancing of the client transactions, for the whole duration of the client sessions, in a Web site comprising a plurality of servers and including a load balancer accessed from a plurality of clients is described. Upon receiving a client initial request the load balancer selects a particular server among the plurality of servers. Then, the initial request is forwarded to the selected server which issues, towards the client, a response uniquely referencing the selected server. Hence, all subsequent requests from the client are forwarded directly to the uniquely referenced server.

[0014] The method of the invention allows to send only the initial request of a client session to the load balancer of a Web site organized as a cluster of servers thus, greatly enhancing the capability of the site to accept new session requests.

[0015] Moreover, the client sessions being effected directly between the client and the server initially selected cannot be later broken by the load balancer.

[0016] Finally, the scheme works regardless of the fact that the client is beyond a proxy or a firewall, on contrary of the previous art, that could only rely on the IP address of the client request to perform load balancing and to decide if a session has ended or not, leading to imperfect results both in terms of load balancing and broken sessions, especially when the actual IP address of the end user is masked by one of the here above mentioned devices.

Brief Description of the Drawings

[0017]

Figure 1 Describes the prior art where a load balancer is dispatching end-user requests over a cluster of servers however, being prevented of always fully taking advantage of the computing resources of the servers when there are too many requests in progress.

Figure 2 Depicts the common case when a Proxy or a Fire-wall is on the way between the end-users and the cluster of servers thus, preventing load balancer to perform a fair dispatching of the incoming requests. Also, the breaking of end-user sessions in case of inactivity is described.

Figure 3 describes the general solution brought by the invention to the shortcomings of the previous art.

Figure 4 describes a particular implementation of the invention insuring always a fair balancing of the workload over all the servers in a cluster of servers and guaranteeing that no session is broken.

Figure 5 describes an alternate implementation of the invention with the same advantages.

Detailed Description of the Preferred Embodiment

[0018] **Figure 1** illustrates the prior art where a load balancer [100] manages to group several servers [105], [110] and [115] together into a cluster [120] that behaves as a single, virtual server so that the service provided is no longer tied to a specific server system. The balanced traffic among servers seems for the end users, like [125], to be a single, virtual server. The site thus appears as a single DNS (Domain Name System) name and IP address to the world [130]. All requests, such as [140], issued from a Web browser, are sent to the IP address of the Load Balancer machine [100], which decides with each client request which server is the best one to accept requests, according to their respective workloads. Hence, load balancer [100] routes the clients' request to the selected server and the server responds directly [135] to the client without any further involvement of the load balancer. In practice, the load balancer receives the IP packets destined to the cluster. These packets have a source and a destination address. The destination address is the IP address of the cluster [130]. All servers in the cluster, i.e. [105], [110] and [115], have their own IP address and know the cluster's IP address too. The dispatcher system checks which server is less busy and routes the packet to that server. The server receives the packet and is able to respond directly to the client based on the source address contained in the packets received by the load balancer. However, with this scheme, all browser requests are always ending up into the load balancer, i.e., the ones of all the sessions in progress plus the ones for the new sessions. If too many requests are converging to the site the bottleneck may become the load balancer itself even though there would be enough computational resources left within the cluster of servers to handle them.

[0019] **Figure 2** illustrates the now frequent case where a sometimes significant portion of a network, for example an intranet [200] shared between a group of related users, is beyond a proxy or a fire-wall or any equivalent device that filters the packets so that the IP addresses of the individual users within the intranet appears to be the same [210] for those that are outside. Then, the load balancer [220] task becomes more difficult to carry out because it tends, after a first request has been received from one of the users on the intranet [200] to keep sending all subsequent ones towards the same server within the cluster [230], for example [240], even though the request is actually coming from a different user on this intranet. Because the proxy [250] has filtered the IP packets from the intranet, load balancer [220] is no longer able to discriminate between the individual users like [260]. In fact, the intranet may be quite large with numerous individual users. If, many of them are accessing the same site, for the same type of service processed by the cluster of servers [230] there is a good

chance that a continuous flow of requests arrives to the load balancer. Then, load balancer is bound to deliver the requests to the initial selected server even if other servers of the cluster, not as busy, could deliver the same service. The above case is not unlikely to happen just because the intranet is shared, e.g., by the personnel of a specific company say, a financial institution. All members may have some interest to consult the same type of information during the day say, stock exchange rates. Therefore, this kind of situation may tend to prevent load balancer to spread an equal share of the workload over all servers of a cluster of servers.

[0020] A second type of problems is encountered if, on contrary of the here above just described situation, no request is arriving for some time to the load balancer so that a significant period of inactivity let think to the load balancer that the user session has ended. Then, it may decide to reassess load balancing with the arrival of another request even though the session is still in progress at the viewpoint of the end-user. This, may happen when a particular end-user is pausing for a long time while the other users on the intranet are not accessing the cluster of server [230]. Therefore, a further request from the end-user [260] may end up in a different server of the cluster, i.e., not in [240]. The new selected server is not aware of the transaction in progress and the context is lost. Hence, a transaction that in progress, e.g., the payment of an item bought from a virtual shop is aborted.

[0021] **Figure 3** depicts the general solution to the problems induced by the use of a load balancer dispatching incoming Web browser requests over a cluster of servers as discussed in the two previous figures. Whenever a new request [300] from a Web browser [305] is issued it is forwarded to the load balancer [310]. This, because it is load balancer DNS name and corresponding IP address (referred to as DNS0 and IP0 [312] in the particular example of figure 3) which is made public for the service or the set of services advertised for the Web site implemented in the form of a cluster of servers [320]. On contrary of the previous art it makes no difference for the invention of receiving the request either directly [326] or through a proxy [325]. Whenever the initial request reaches the load balancer [311] it is dispatched to one of the servers of the cluster of servers [320]. The decision of routing towards one particular server, like [313] in this example, is the prime job of the load balancer. The metric used to decide which server is to be selected at a given instant depends on the design of the load balancer which is assumed to collect from all the servers, at regular intervals, performance information regarding their level of activity. In broad general terms it can be said that the less busy of the servers is selected in an attempt to indeed reach the goal of balancing the workload equally over all the servers. The invention does not, per se, interferes with this process which is under the sole responsibility of the load balancer. However, it contributes dramatically to improve the job of the load balancer by forwarding to it only the initial requests, like [300], issued by the Web browser [305] when initializing a session as this will become apparent in the following. Thus, the request is forwarded, in the example of figure 3, to server [321] that met the criterion for being elected to process initial request [311] at time it reached the load balancer [310] on the basis of the performance data that were collected by the load balancer from the servers. At this point the rest of the session is going to be handled solely by the particular server, e.g., [321] without any further implication of the load balancer which is then completely free to accept all new requests arriving to the site and generally referred to as "hits" in the literature that deals with Web site performances, at least until not all the resources of the cluster of servers are completely exhausted. This is a complete departure from the prior art where the processing of the new hits interfered with the processing of all the sessions already in progress as illustrated in previous figures thus, leading to postpone the processing of a new request by the load balancer when it is too busy itself dispatching the numerous requests of the sessions already in progress, even though there may have still plenty of computing resources available hence, wasted in the cluster of servers. The above is made possible because each of the servers within the cluster of servers has its own unique DNS name et corresponding IP address. For example DNS1 and IP1 for the server [321]. Therefore, the server that has been elected by the load balancer, upon receiving the initial request [313] will reply directly [330] to the Web browser of the end-user. This reply mentions the Uniform Resource Locator or URL to be used for the further requests of that session. This URL contains the DNS name or the IP address of the elected server i.e.: DNS1 or IP1. All further requests [332] are forwarded directly to the selected server thus, freeing the load balancer of dispatching the subsequent session requests as mentioned above and insuring that all are going to reach the same server for the whole duration of the session. Again, the Web browser and the cluster of servers may be on either side of a proxy like [325] without impacting, at all, the above scheme on contrary of the previous art.

[0022] **Figure 4** illustrates one particular implementation of the general solution depicted in figure 3. It takes advantage of the here above mentioned option that a specific response could be issued by the selected server for informing the end-user browser of the actual DNS name of the server elected to process its session. The protocol, part of the TCP/IP suite of protocols, used by the Web server to transfer hypermedia documents across the Internet, known under the acronym of HTTP for Hyper Text Transport Protocol, specifically foresee the possibility of redirecting a request that was issued for a specific DNS name to another DNS name for the duration of the session. Hence, whenever the selected server [400] is receiving a new request [410] it has just to respond with a HTTP redirect command [420], either

directly or through a proxy [425], destined to the end-user browser [430] and carrying the actual DNS name of the server in charge [400] so that the rest of the session is going to take place, as required, directly between the remote browser and the particular server.

[0023] **Figure 5** is another example of how to implement the general solution of figure 3 so that to solve the shortcomings of the prior art. In this approach the "WELCOME" page [500] of every server, within the cluster of servers, supporting a given set of services, is identical except for the DSN names [511] or [512] explicitly referencing the particular DSN address of the server on which "WELCOME" page is loaded. It is worth noting here that if , for the sake of clarity, this particular example is illustrated with only two servers, i.e., Server_1 [501] and Server_2 [502] it must obviously be understood that any number of servers could be considered instead and solution still applies. Thus, it is load balancer responsibility to decide which server is going to process a new arriving request from a remote end-user browser by directing it either to Server_1 [501] or Server_2 [502]. Whichever is elected it forwards its own "WELCOME" page [500] to the end-user. Then, when the end-user decide to use "FOOBAR", a service offered at this site and displayed in the "WELCOME" page menu, a new request is sent by the browser which address directly Server_1 or Server_2 depending on the DSN address that was contained in the received "WELCOME" page i.e. ../dsn1/.. [511] or ../dsn2/.. [512]. From that point on the other pages of the "FOOBAR" service are referenced relatively to the first page without any further reference to the specific server (using relative URL's or Uniform Resource Locator). That is, only the link is specified as shown in [521] and [522]. Therefore, all further requests from the end-user browser are reaching directly the server that was chosen by the load balancer at the beginning of the session allowing to fully carry out the solution described in general terms in figure 3 thus, enabling all of its advantages.

Claims

1. A method for preserving load balancing of the client transactions, for the whole duration of the client sessions, in a Web site comprising a plurality of servers and including a load balancer accessed from a plurality of clients, said method comprising the steps of:

   receiving a client initial request by said load balancer; and

   selecting from said load balancer a particular server among said plurality of servers; and

   forwarding said initial request to said selected server; and

   issuing from said selected server, towards said client, a response uniquely referencing said selected server; and

   forwarding directly all subsequent requests from said client to said uniquely referenced server.

2. The method according to claim 1 wherein:
   said response from said selected server consists, before said client initial request is honored, in issuing a redirection command to said client including the unique reference of said selected server.

3. The method according to claim 1 wherein:
   said response from said selected server self-contains said unique reference of said selected server.

4. The method according to any one of the previous claims wherein:
   said client initial request is the only request received by said load balancer from said client for the whole duration of said client session.

5. The method according to any one of the previous claims wherein:
   said client session is processed by said selected server until said session is ended by said client.

6. A system, in particular a Web site comprising a plurality of servers and including a load balancer, comprising means adapted for carrying out the method according to any one of the previous claims.

7. A computer-like readable medium comprising instructions for carrying out the method according to any one of the claims 1 to 5.

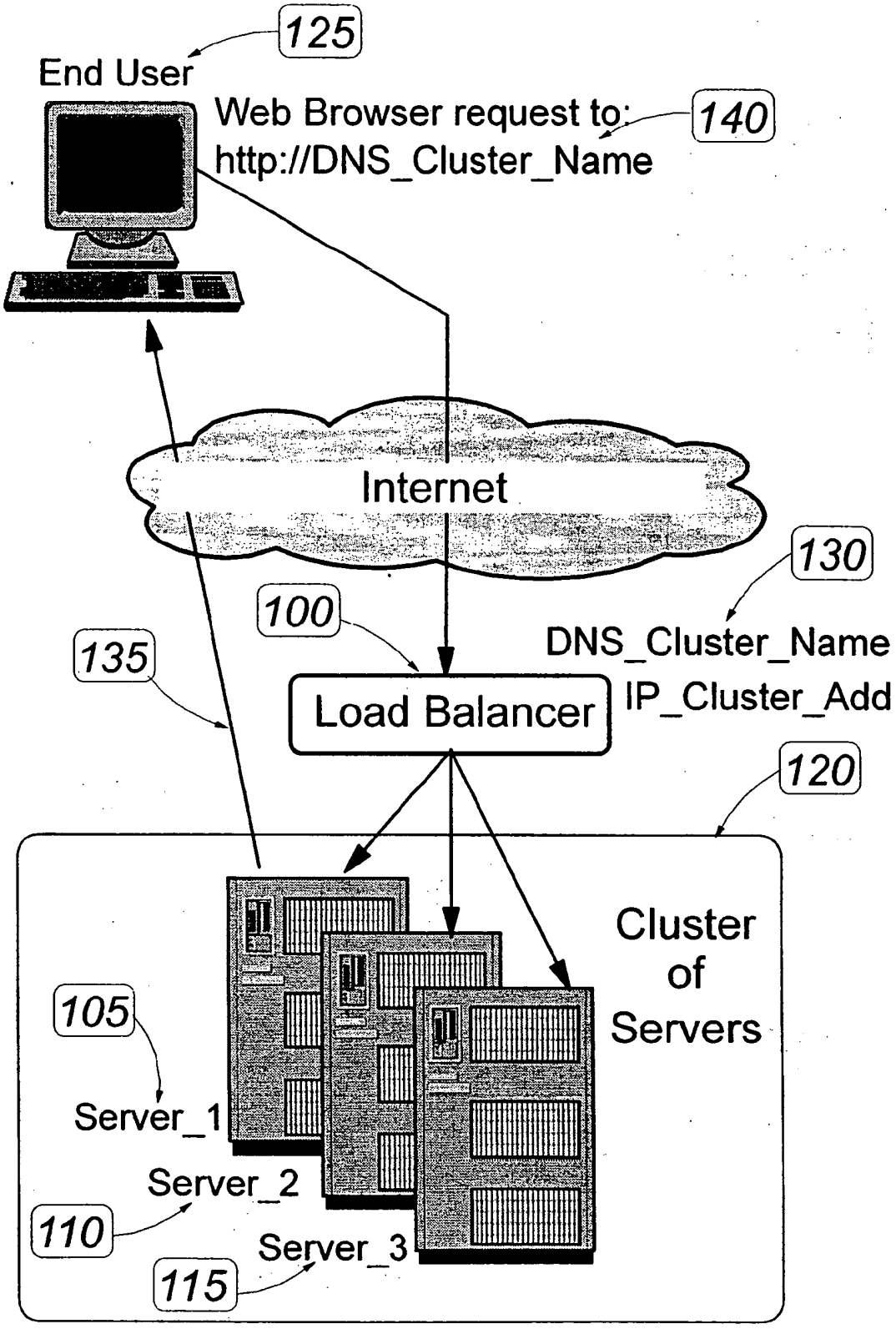
125
End User
Web Browser request to:
http://DNS_Cluster_Name
140
Internet
130
DNS_Cluster_Name
IP_Cluster_Add
100
135
Load Balancer
120
Cluster
of
Servers
105
Server_1
Server_2
110
Server_3
115

Figure 1

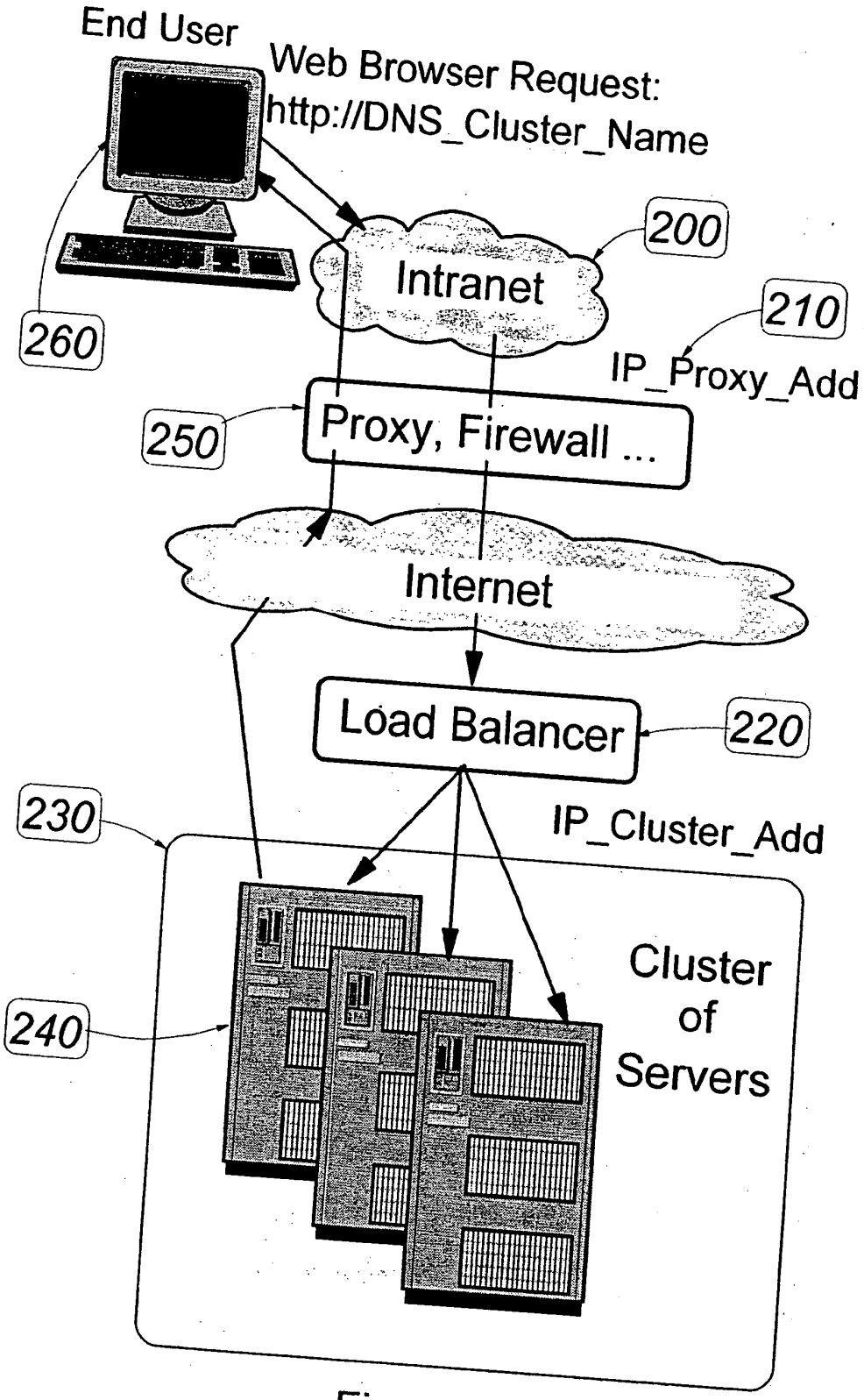
End User
Web Browser Request:
http://DNS_Cluster_Name
200
Intranet
210
IP_Proxy_Add
260
250
Proxy, Firewall ...
Internet
Load Balancer
220
230
IP_Cluster_Add
240
Cluster
of
Servers

Figure 2

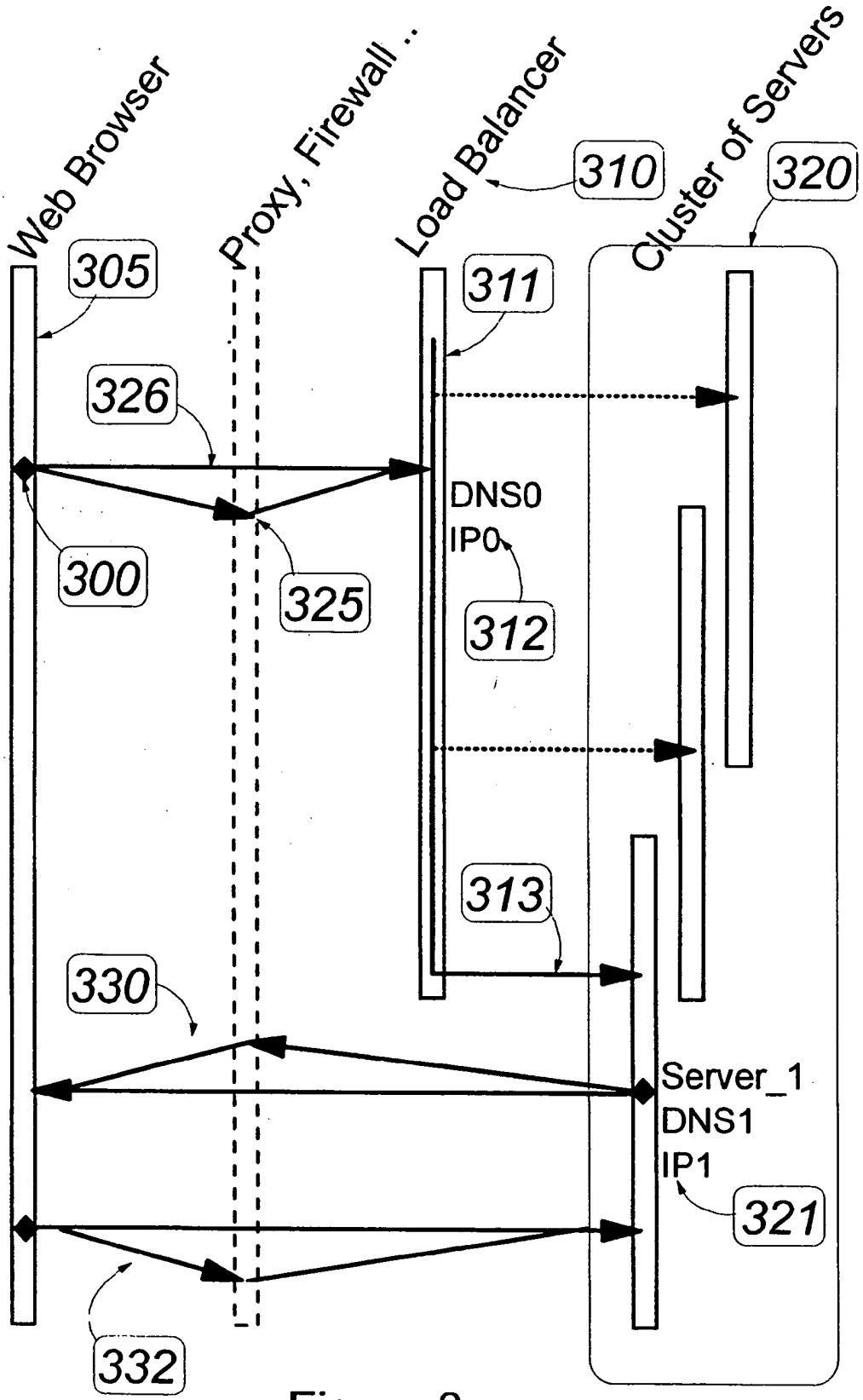
Web Browser
Proxy, Firewall ..
Load Balancer
Cluster of Servers
310
320
305
311
326
DNS0
IP0
300
325
312
313
330
Server_1
DNS1
IP1
321
332

Figure 3

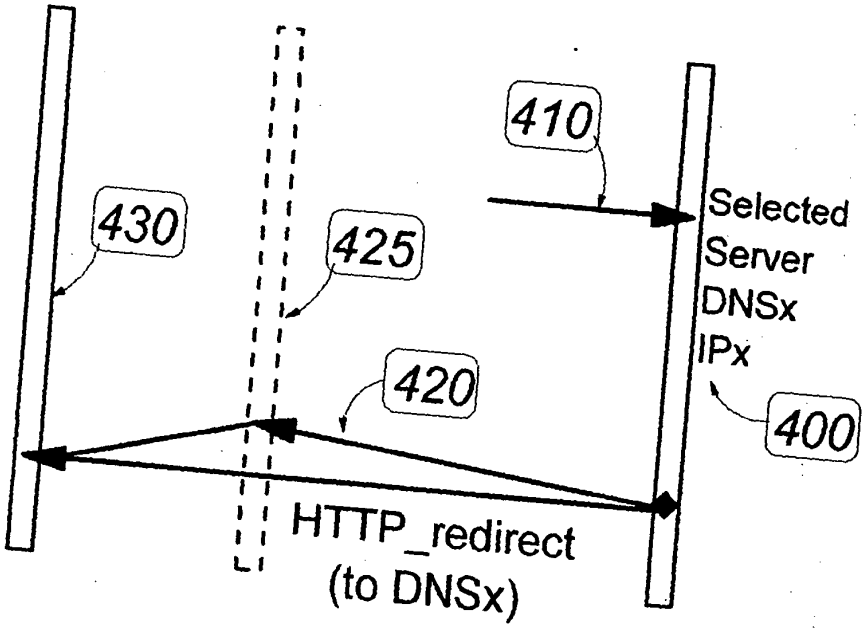
410
430
425
Selected
Server
DNSx
IPx
420
400
HTTP_redirect
(to DNSx)

Figure 4

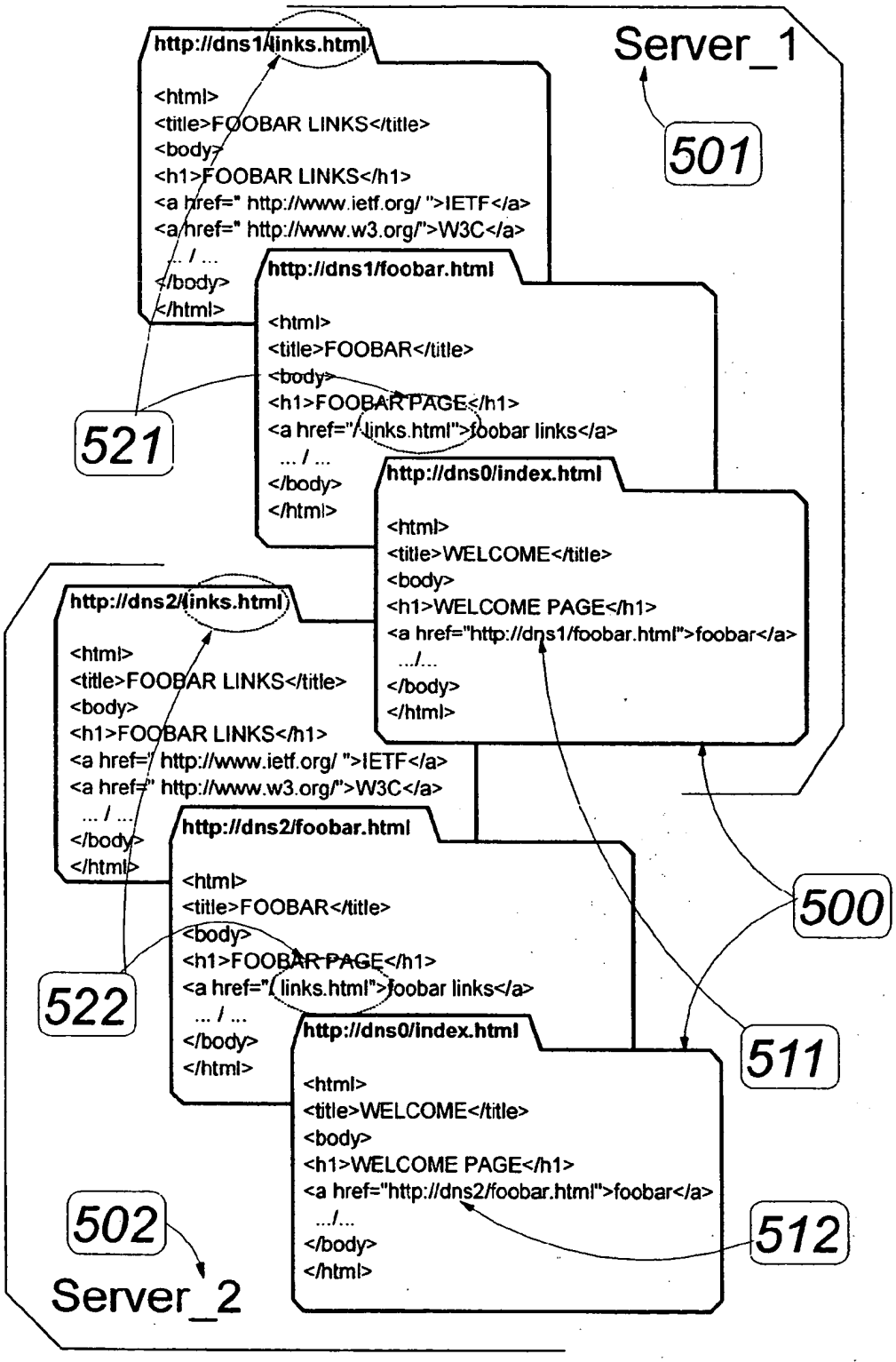
Server_1
501
http://dns1/links.html
<html>
<title>FOOBAR LINKS</title>
<body>
<h1>FOOBAR LINKS</h1>
IETF
W3C
... / ...
</body>
</html>
http://dns1/foobar.html
<html>
<title>FOOBAR</title>
<body>
<h1>FOOBAR PAGE</h1>
foobar links
... / ...
</body>
</html>
http://dns0/index.html
<html>
<title>WELCOME</title>
<body>
<h1>WELCOME PAGE</h1>
foobar
.../...
</body>
</html>
521
http://dns2/links.html
<html>
<title>FOOBAR LINKS</title>
<body>
<h1>FOOBAR LINKS</h1>
IETF
W3C
... / ...
</body>
</html>
http://dns2/foobar.html
<html>
<title>FOOBAR</title>
<body>
<h1>FOOBAR PAGE</h1>
foobar links
... / ...
</body>
</html>
http://dns0/index.html
<html>
<title>WELCOME</title>
<body>
<h1>WELCOME PAGE</h1>
foobar
.../...
</body>
</html>
522
500
511
512
502
Server_2

Figure 5

European Patent Office

EUROPEAN SEARCH REPORT

Application Number
EP 99 48 0027

DOCUMENTS CONSIDERED TO BE RELEVANT

Category	Citation of document with indication, where appropriate, of relevant passages	Relevant to claim	CLASSIFICATION OF THE APPLICATION (Int.Cl.7)
Y	"METHOD FOR DYNAMICALLY ROUTING WEB REQUESTS TO DIFFERENT WEB SERVERS" IBM TECHNICAL DISCLOSURE BULLETIN, vol. 40, no. 12, 1 December 1997 (1997-12-01), pages 5-8, XP000754080 ISSN: 0018-8689 * the whole document *	1-7	H04L29/06
Y	US 5 867 706 A (MARTIN SEAN JAMES ET AL) 2 February 1999 (1999-02-02) * column 5, line 7 - column 7, line 17 *	1-7	
A	MOURAD A ET AL: "SCALABLE WEB SERVER ARCHITECTURES" PROCEEDINGS IEEE SYMPOSIUM ON COMPUTERS AND COMMUNICATIONS, 1 July 1997 (1997-07-01), pages 12-16, XP000199852 * page 12, left-hand column, paragraph 2 - page 14, right-hand column, paragraph 1 *	1-3	
A	ANDERSEN D ET AL: "SWEB: towards a scalable World Wide Web server on multicomputers" PROCEEDINGS OF THE INTERNATIONAL PARALLEL PROCESSING SYMPOSIUM, 15 April 1996 (1996-04-15), pages 850-856, XP002088154 * page 851, right-hand column, paragraph 5 - page 852, left-hand column, paragraph 3 *	1-3	TECHNICAL FIELDS SEARCHED (Int.Cl.7) H04L G06F

The present search report has been drawn up for all claims

Place of search	Date of completion of the search	Examiner
THE HAGUE	8 October 1999	RAMIREZ DE AREL.., F

CATEGORY OF CITED DOCUMENTS

X : particularly relevant if taken alone
Y : particularly relevant if combined with another document of the same category
A : technological background
O : non-written disclosure
P : intermediate document

T : theory or principle underlying the invention
E : earlier patent document, but published on, or after the filing date
D : document cited in the application
L : document cited for other reasons

& : member of the same patent family, corresponding document

EPO FORM 1503 03.82 (P04C01)

ANNEX TO THE EUROPEAN SEARCH REPORT ON EUROPEAN PATENT APPLICATION NO. EP 99 48 0027

This annex lists the patent family members relating to the patent documents cited in the above-mentioned European search report. The members are as contained in the European Patent Office EDP file on
The European Patent Office is in no way liable for these particulars which are merely given for the purpose of information.

08-10-1999

Patent document cited in search report	Publication date	Patent family member(s)	Publication date
US 5867706 A	02-02-1999	GB 2309558 A	30-07-1997
		CN 1202971 A	23-12-1998
		CZ 9802324 A	16-12-1998
		EP 0880739 A	02-12-1997
		WO 9729423 A	14-08-1997
		JP 11503551 T	26-03-1999
		PL 327918 A	04-01-1999
		DE 69602461 D	17-06-1999
		ES 2131415 T	16-07-1999

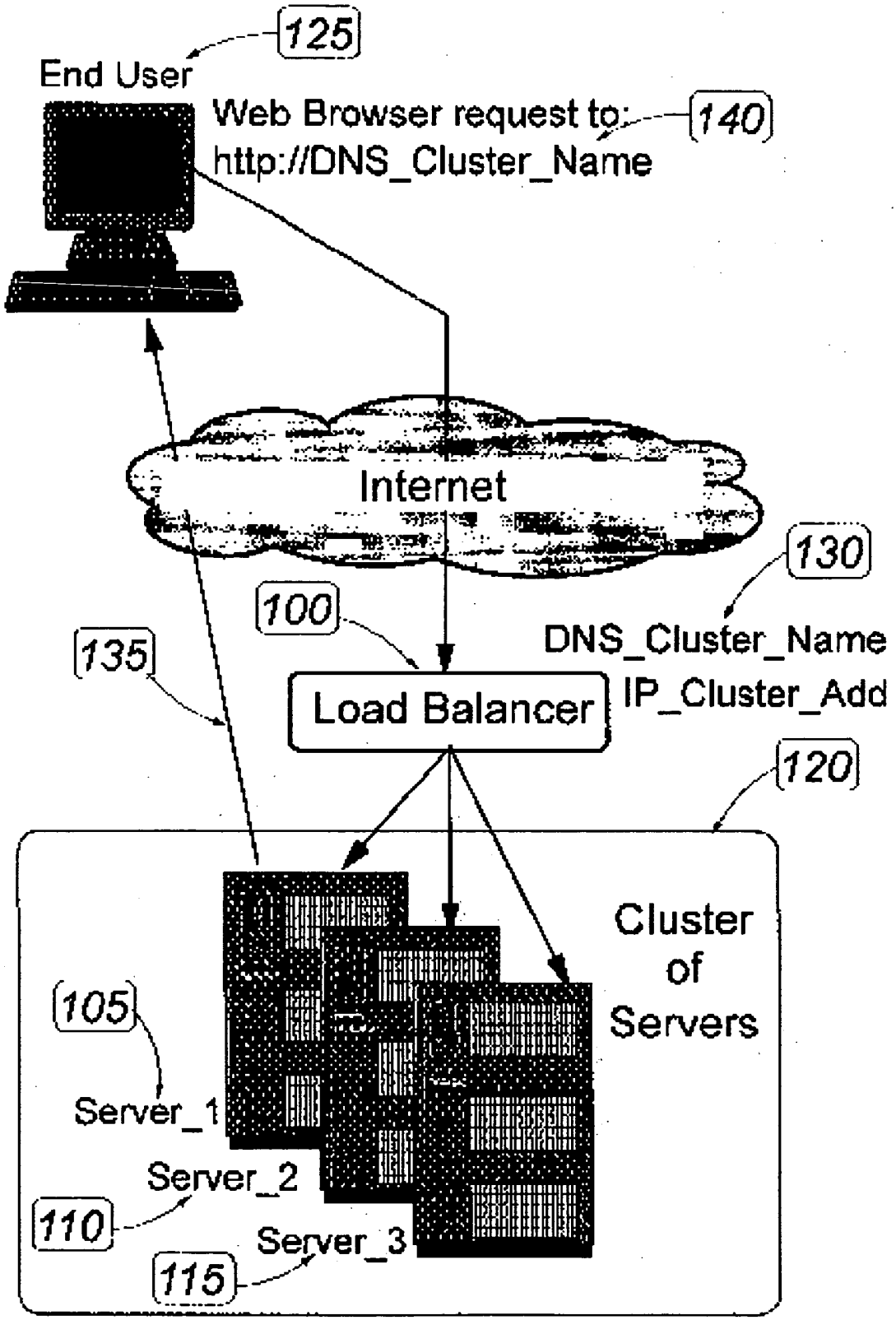
125
End User
Web Browser request to:
http://DNS_Cluster_Name
140
Internet
130
DNS_Cluster_Name
IP_Cluster_Add
100
135
Load Balancer
120
Cluster
of
Servers
105
Server_1
Server_2
110
Server_3
115

Figure 1

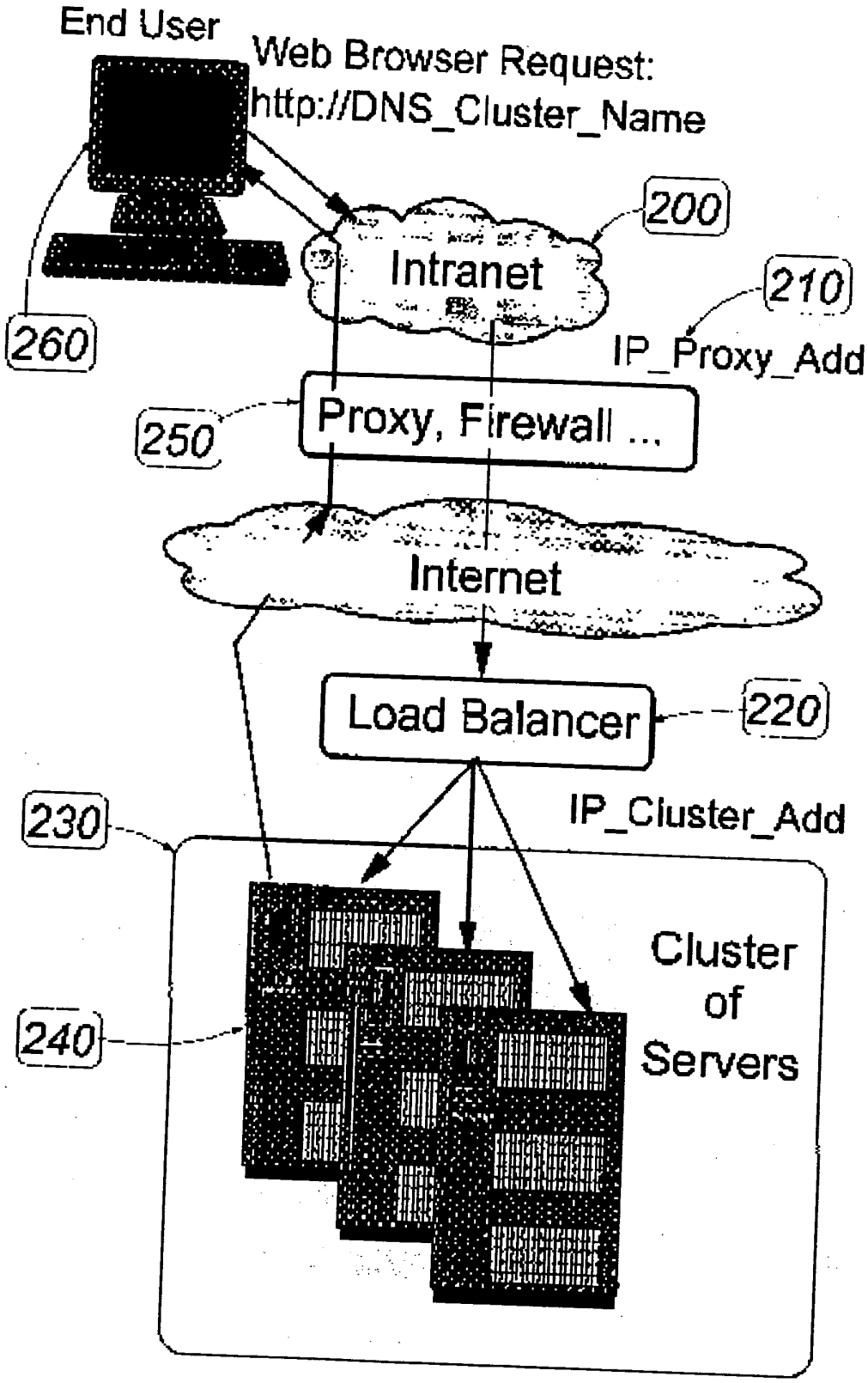
End User
Web Browser Request:
http://DNS_Cluster_Name
200
Intranet
210
IP_Proxy_Add
260
250
Proxy, Firewall ...
Internet
Load Balancer
220
230
IP_Cluster_Add
Cluster
of
Servers
240

Figure 2

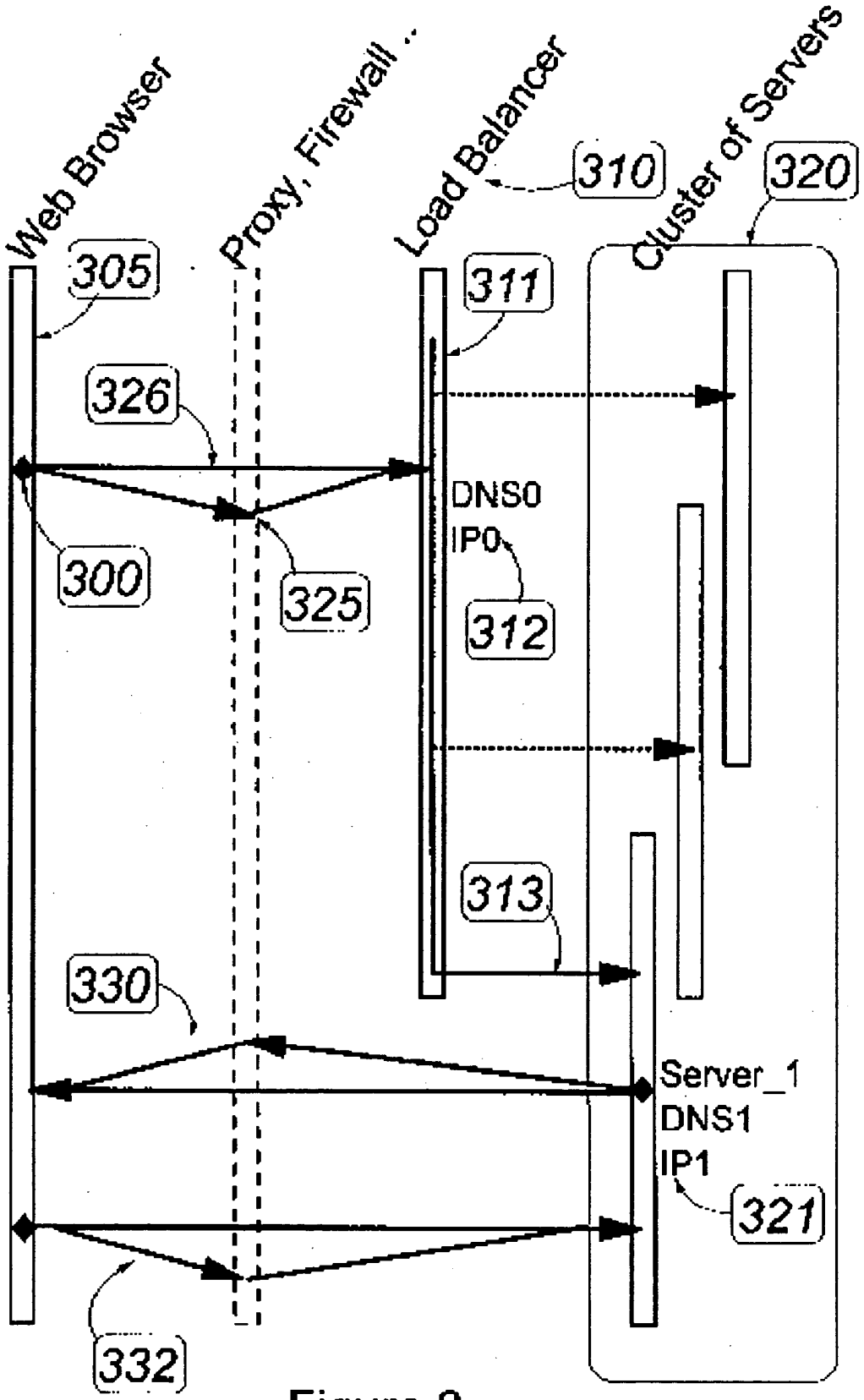
Web Browser
Proxy, Firewall ..
Load Balancer
Cluster of Servers
310
320
305
311
326
DNS0
IP0
300
325
312
313
330
Server_1
DNS1
IP1
321
332

Figure 3

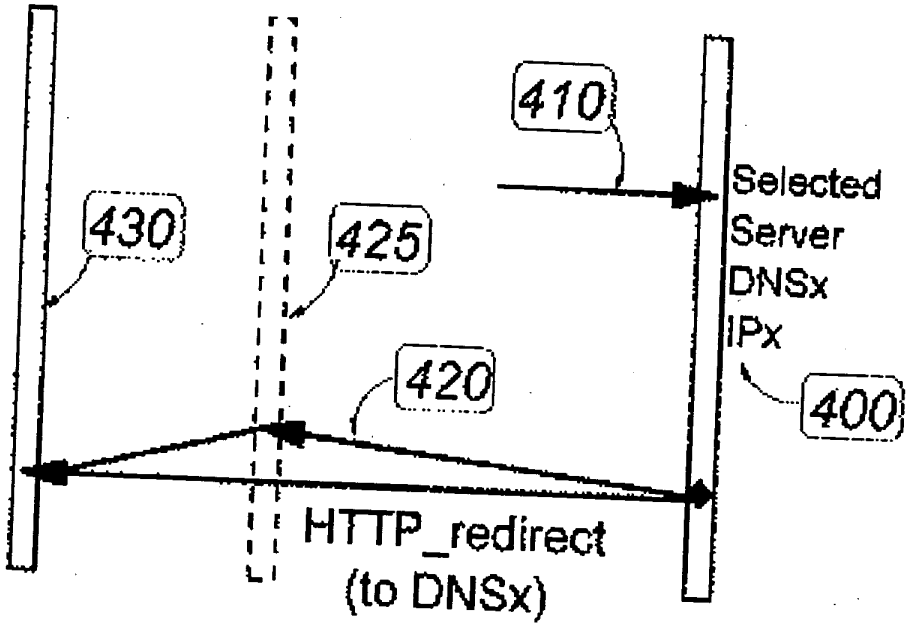
410
Selected
Server
DNSx
IPx
400
430
425
420
HTTP_redirect
(to DNSx)

Figure 4

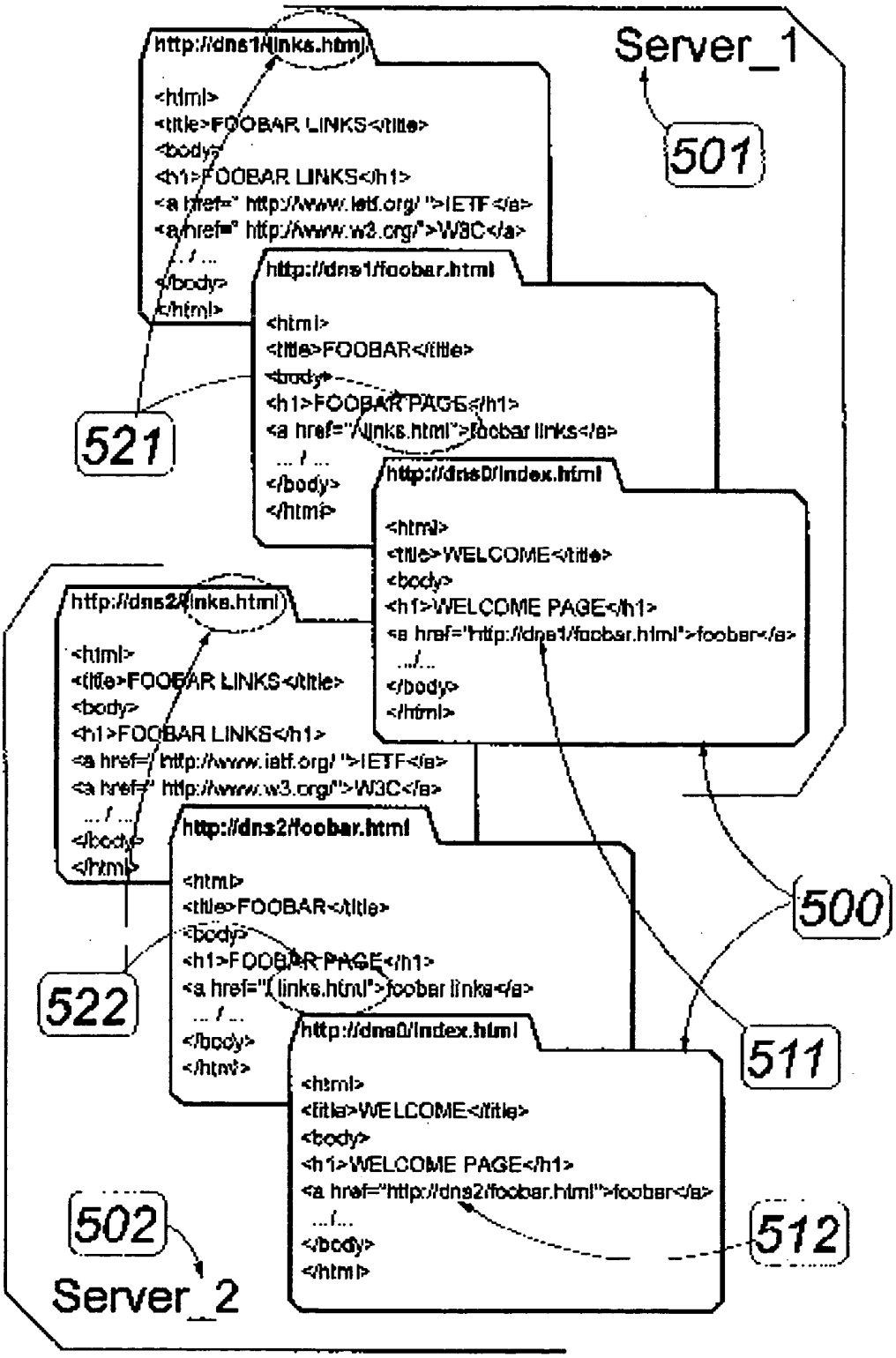
Server_1
501
http://dns1/links.html
<html>
<title>FOOBAR LINKS</title>
<body>
<h1>FOOBAR LINKS</h1>
IETF
W3C
... / ...
</body>
</html>
http://dns1/foobar.html
<html>
<title>FOOBAR</title>
<body>
<h1>FOOBAR PAGE</h1>
foobar links
... / ...
</body>
</html>
521
http://dns0/index.html
<html>
<title>WELCOME</title>
<body>
<h1>WELCOME PAGE</h1>
foobar
.../...
</body>
</html>
http://dns2/links.html
<html>
<title>FOOBAR LINKS</title>
<body>
<h1>FOOBAR LINKS</h1>
IETF
W3C
... / ...
</body>
</html>
http://dns2/foobar.html
<html>
<title>FOOBAR</title>
<body>
<h1>FOOBAR PAGE</h1>
foobar links
... / ...
</body>
</html>
522
http://dns0/index.html
<html>
<title>WELCOME</title>
<body>
<h1>WELCOME PAGE</h1>
foobar
.../...
</body>
</html>
500
511
512
502
Server_2

Figure 5